AMENDMENTS TO THE CLAIMS

For the convenience of the Examiner, all claims have been presented whether or not an amendment has been made. The claims have been amended as follows:

1. **(Currently amended)** A system, comprising:

a finite state machine operating within a portable thread environment wherein a plurality of threads communicate with each other **while cooperatively completing a task**; and

a plurality of threads operable to cooperatively complete a task and ~~one or more PTE message generators~~ configured to pass event information **associated with the task** ~~contained~~ in PTE messages to the finite state machine, wherein the finite state machine changes ~~states according to~~ **a state associated with the task based on** the event information.

2. **(Currently amended)** The system of claim 1, wherein the event information comprises one or more events passed to a thread and **the state associated with the task** ~~a present state of the finite state machine~~.

3. **(Original)** The system of claim 2, wherein the finite state machine comprises:

a message interpreter configured to accept the PTE messages; wherein the interpreter maps the messages to actions using the look-up table.

4. **(Previously Presented)** The system of claim 3, wherein the finite state machine further comprises:

a storage device for storing the one or more actions, said actions used to generate PTE messages.

5. **(Currently amended)** The system of claim 4, wherein the finite state machine further comprises:

a state changer configured to change the state **associated with the task** ~~of the finite state machine~~ based upon event information and **a previous** ~~the previous~~ state **associated with the task** ~~of the finite state machine~~.

6. **(Currently amended)** A method comprising:

receiving**, at a finite state machine,** PTE messages **from a plurality of threads cooperatively completing a task** ~~**by a finite state machine**~~ in a portable thread environment, wherein the messages contain event information comprising **a present state associated with the task** ~~**the present state of the finite state machine**~~;

mapping the **event** ~~**state transition**~~ information **and the present state to one or more actions** ~~**with actions**~~ stored in a storage device; and

changing **a state associated with the task** from **the present** ~~**a first**~~ state to a second state based upon the **present** ~~**first**~~ state and the event information.

7. **(Currently amended)** The method of claim 6, wherein **the task remains associated with the present state** ~~**the finite state machine stays in the first state**~~ based upon the **present** ~~**first**~~ state and the actions.

8. **(Currently amended)** The method of claim 7, further comprising:

generating state machine events relating to the state **associated with the task** ~~**of the finite state machine**~~.

9. **(Original)** The method of claim 8, further comprising:

distributing the state machine events between one or more threads in the portable thread environment.

10. **(Original)** The method as in claim 9, further comprising:

distributing the state machine events between one or more threads in the portable thread environment and a second portable thread environment.

11. **(Currently amended)** A system, comprising:

means for receiving**, at a finite state machine,** PTE messages **from a plurality of threads cooperatively completing a task** ~~by a finite state machine~~ in a portable thread environment, wherein the messages contain event information comprising **a present state associated with the task** ~~the present state of the finite state machine~~;

means for mapping the event information **and the present state to one or more actions** ~~with actions~~ stored in a storage device; and

means for changing **a state associated with the task** from **the present** ~~a first~~ state to a second state based upon the **present** ~~first~~ state and the event **information**.

12. **(Currently amended)** The system of claim 11, wherein the **task remains associated with the present state** ~~finite state machine stays in the first state~~ based upon the **present** ~~first~~ state and the event **information**.

13. **(Currently amended)** The system of claim 12, further comprising:

means for generating state machine events relating to **the state associated with the task** ~~a state of the finite state machine~~.

14. **(Original)** The system of claim 13, further comprising:

means for distributing the state machine events between one or more threads in the portable thread environment.

15. **(Original)** The system of claim 14, further comprising:

means for distributing the state machine events between one or more threads in the portable thread environment and a second portable thread environment.

16. **(Currently amended)** A computer-readable medium having stored thereon a plurality of instructions, said plurality of instructions when executed by a computer, cause said computer to perform:

receiving**, at a finite state machine,** PTE messages **from a plurality of threads cooperatively completing a task** ~~by a finite state machine~~ in a portable thread environment, wherein the messages contain event information comprising **a present state associated with the task** ~~the present state of the finite state machine~~;

mapping the event information **and the present state to one or more actions** ~~with actions~~ stored in a storage device; and

changing **a state associated with the task** from **the present** ~~a first~~ state to a second state based upon the **present** ~~first~~ state and the event **information**.

17. **(Currently amended)** The computer-readable medium of claim 16, wherein **the task remains associated with the present state** ~~the finite state machine stays in the first state~~ based upon the first state and the events.

18. **(Currently amended)** The computer-readable medium of claim 17 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:

generating state machine events relating to **the state associated with the task** ~~a state of the finite state machine~~.

19. **(Original)** The computer-readable medium of claim 18 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:

distributing the state machine events between one or more threads in the portable thread environment.

20. **(Original)** The computer-readable medium of claim 19 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:

distributing the state machine events between one or more threads in the portable thread environment and a second portable thread environment.